



Attorney's Docket No.: 09765-020001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : James N. Rees

Art Unit : 2143

Serial No. : 09/704,221

Examiner : P. Nguyen

Filed : November 1, 2000

Title : TRANSMITTING INFORMATION OVER A NETWORK

Mail Stop Appeal Brief - Patents

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

BRIEF ON APPEAL

The Appeal Brief fee of \$500 is enclosed. Please apply any other charges or credits to Deposit Account No. 06-1050.

02/23/2005 HDEMESS1 00000026 09704221

01 FC:1402

500.00 OP

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

I hereby certify under 37 CFR §1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date of Deposit

Signature

Typed or Printed Name of Person Signing Certificate

(i.) Real Party In Interest

The real party in interest in the above application is ITA Software, Inc.

(ii.) Related Appeals and Interferences

The appellant is not aware of any appeals or interferences related to the above-identified patent application.

(iii.) Status of Claims

This is an appeal from the decision of the Primary Examiner in an Office Action dated September 2, 2004, rejecting claims 1-21, all of the claims of the above application.

(iv.) Status of Amendments

Appellant filed a Reply to the final action on November 18, 2004. Appellant amended claims 1, 10 and 14 to correct the tense of a verb. The examiner has not apprised Appellant as to whether the examiner entered the amendments. Appellant filed a Notice of Appeal on November 18, 2004.

(v.) Summary of Claimed Subject Matter

Background

This invention relates to transmitting information over a network.

Network servers, such as web servers, normally receive requests for information from remote client computers over a network, such as the Internet or an intranet. Network servers may transmit information to the client computer in response to a request. It is sometimes necessary to transmit an executable to the client computer along with data, because the executable is used to process the data on the client. For example, a web server may respond to a request for flight itineraries by sending flight itinerary data along with a JAVATM applet to allow the client to graphically display the data.

Appellant's Invention

Claim 1

Claim 1 is directed to a method. Claim 1 includes the features of transmitting information by transmitting an executable from a server to a remote location over a network, the executable processing data generated by the server. FIG. 1 shows a server 10 receives requests 12a, 12b for travel information 18 from remote client computers 14a, 14b over a network 16. The server 10 responds to the request 12 by transmitting travel data and a JAVA™ applet to the client over the network 16. The applet is executed on the client computer to process the travel data. (Appellant's specification Page 3, lines 14-18).

A web server program 28 receives the requests 12 from a client computers 14a, 14b through the network interface 20 and responds to the requests by transmitting information 18 through the network interface 20 to the client computer. Web server 28 may send web pages 30 or a client executable 32, such as a JAVA™ applet. (Appellant's specification Page 3, lines 26-30).

Claim 1 also requires generating the data at the server as the executable is being transmitted. The server 10 is configured to generate the travel data while transmitting the applet and then later transmit the travel data to the client computer, as will be described in greater detail below. (Appellant's specification Page 3 lines 18-20) "The web server 28 transmits at least part of client executable 32 while the travel module is generating the segment of the web page." (Appellant's specification Page 5, lines 28-30).

Claim 1 also requires transmitting the generated data to the executable at the remote location over the network, the executable processing the data by decoding and uncompressing the data for graphical presentation on a display associated with the remote location and displaying the decoded and uncompressed data in a graphical presentation on a display device associated with the remote location.

"The server 10 is configured to generate the travel data while transmitting the applet and then later transmit the travel data to the client computer, as will be described in greater detail below." As shown in FIG. 4, client executable 32 presents a panel 70 on a web page 30c that informs 72 the user that the executable 32 is waiting for data 36 on travel itineraries associated

with the user's request 12. When web browser 42 receives the data 36, client executable 32 processes the data, for example, by decoding, uncompressing, and preparing the data 36 for graphical presentation on the display 50. (Appellant's specification Page 5, lines 13-17).

Claim 10

Claim 10 embodies the invention as generally set forth in claim 1 as an article comprising a machine-readable medium which stores machine-executable instructions operable to cause a machine to perform recited functions. The server 10 also includes a computer-readable storage subsystem 24 that stores computer programs, which are executed by processor 22. (Appellant's specification Page 3, lines 22-25). The functions performed by the computer program product are the equivalent to the method actions discussed above.

Claim 14

Claim 14 expresses the invention generally set forth in claim 1 as an apparatus. The server 10 also includes a computer-readable storage subsystem 24 that stores computer programs, which are executed by processor 22. (Appellant's specification Page 3, lines 22-25) The functions performed by the apparatus are the equivalent to the method actions discussed above.

(vi.) Grounds of Rejection to be Reviewed on Appeal

(1) Claims 1-21 stand rejected 35 U.S.C. 102(e), as being obvious over House, et al. U.S. Patent 6,188,400 in view of Tso et al., U.S. Patent 6,421,733.

(vii.) Argument

Obviousness

"It is well established that the burden is on the PTO to establish a *prima facie* showing of obviousness, *In re Fritsch*, 972 F.2d. 1260, 23 U.S.P.Q.2d 1780 (C.C.P.A., 1972)."

"It is well established that there must be some logical reason apparent from the evidence or record to justify combination or modification of references. *In re Regal*, 526 F.2d 1399 188, U.S.P.Q.2d 136 (C.C.P.A. 1975). In addition, even if all of the elements of claims are disclosed

in various prior art references, the claimed invention taken as a whole cannot be said to be obvious without some reason given in the prior art why one of ordinary skill in the art would have been prompted to combine the teachings of the references to arrive at the claimed invention. *Id.* Even if the cited references show the various elements suggested by the Examiner in order to support a conclusion that it would have been obvious to combine the cited references, the references must either expressly or impliedly suggest the claimed combination or the Examiner must present a convincing line of reasoning as to why one skilled in the art would have found the claimed invention obvious in light of the teachings of the references. *Ex Parte Clapp*, 227 U.S.P.Q.2d 972, 973 (Board. Pat. App. & Inf. 985)."

"The mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification." *In re Gordon*, 221 U.S.P.Q. 1125, 1127 (Fed. Cir. 1984).

Although the Commissioner suggests that [the structure in the primary prior art reference] could readily be modified to form the [claimed] structure, "[t]he mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification." *In re Laskowski*, 10 U.S.P.Q. 2d 1397, 1398 (Fed. Cir. 1989).

"The claimed invention must be considered as a whole, and the question is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination." *Lindemann Maschinenfabrik GMBH v. American Hoist & Derrick*, 221 U.S.P.Q. 481, 488 (Fed. Cir. 1984).

Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion supporting the combination. Under Section 103, teachings of references can be combined only if there is some suggestion or incentive to do so. *ACS Hospital Systems, Inc. v. Montefiore Hospital*, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984) (emphasis in original, footnotes omitted).

"The critical inquiry is whether 'there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination.'" *Fromson v. Advance Offset Plate, Inc.*, 225 U.S.P.Q. 26, 31 (Fed. Cir. 1985).

**(1) Claims 1-21 are allowable over House, et al.
U.S. Patent 6,188,400 in view of Tso et al., U.S.
Patent 6,421,733.**

Claims 1, 2, 10, 14, and 15

Claim 1 includes the features of transmitting an executable from a server to a remote location ... the executable processing data generated by the server and generating the data at the server as the executable is being transmitted and transmitting the generated data to the executable at the ... client. Claim 1 also includes the feature that the executable processes the data by decoding and uncompressing the data for graphical presentation on a display associated with the remote location. House, in view of Tso, does not disclose this combination of features of claim 1.

The examiner contends that House teaches (at pages 2 and 3 of the final office action):

... transmitting an executable e.g., Java Applet from a server to a remote location (eg. client 102 of Figure 1) over a network (eg communication between client 102 and network server 110) the executable processing data generated by the server (Col. 6 lines 23-30); generating the data, as the executable is being transmitted, and transmitting the generated data to the executable at the remote location over the network (eg. server generate output data based on the first applet send from the client 101 and send the output data to the client 101) Figure 5b, and col. 7 last paragraph through col. 8 1st paragraph), the executable processing the data by displaying the data in a graphical presentation on a display device associated with the remote location (Abstract col. 8 lines 6-10) (Sic)

The examiner relies on Tso to teach "decoding and decompressing the data for display." (Office action page 3, lines 8-9).

House describes a process summarized in Figure 5B in which control information is generated by an applet in a browser 108 on a client 102. The control information is transmitted

from the client 102 to the network server 110. A script is executed on the network server 110 using the control information. The script produces data that is output from the network server to the browser 108, where if desired, the data can be displayed on the browser 108 or used to execute 518 an applet in the browser 108. Specifically, House at Col 7 line 65 to Col. 8 line 10 discloses:

FIG. 5B is a flow chart illustrating the method steps employed in implementing the foregoing. First, control information is generated by an applet in the browser 108. This is illustrated in block 510. That control information is transmitted from the client 102 to the network server 110 and received 512 therein. Next, using this control information, a script is executed on the network server 110. This is represented by block 514. Execution can be initiated either by the receipt of the control information, or by other means. Next, as shown in block 516, the resulting script output data is then transmitted from the network server to the browser 108, where if desired, the data can be displayed (sic) on the browser 108 or used to execute 518 an applet in the browser 108.

Thus, House, as disclosed in passage quoted above, describes a process in which control information generated by the client invokes a script on the server, which sends output data back to the client. House teaches that the applet produces control information to invoke the executable on the server.

In contrast, Appellant's claim 1 requires transmitting an executable from a server to a remote location . . . , with the executable processing data generated by the server and generating the data at the server as the executable is being transmitted.

In the examiner's action (page 2 paragraph 5), the examiner contends that House teaches "transmitting an executable (e.g. java applet) (sic) from a server to a remote location." The examiner further contends that Col. 6, lines 23-30 teach that the applet is transmitted from a server to the client, the executable processing data generated by the server. This passage from House is set forth below:

A Java applet is created for each scriptable control provided by the present invention. These applets may be customized by the

developer. Since there is one applet per control, if there are five command buttons (for example) on one form, only one copy of the command button applet is downloaded to the client browser 108. This applet remains in the browser cache.

However, this passage does not disclose that the applet is transmitted from the server that is generating the data. The passage merely discloses that the applet is downloaded. Where it is downloaded from is not discussed in House.

Nevertheless, Appellant's Claim 1 additionally requires the feature of "generating the data at the server as the executable is being transmitted." House clearly does not suggest this feature, whether the executable is seen as being on the server as discussed above, or the Applet as contended by the examiner. For example, in the passage quoted above, House teaches that "A Java applet is created for each scriptable control provided by the present invention." House also states that "... only one copy of the command button applet is downloaded to the client browser 108. This applet remains in the browser cache." Since House, clearly states that the applet remains in the browser cache, the applet is not downloaded while the data is generated, because unlike Appellant's claim 1, House requires execution of the applet on the client system in order to generate the data.

Appellant's feature of "generating the data at the server as the executable is being transmitted" is an important feature of Appellant's invention and is not suggested by House or Tso. Appellant notes that it is sometimes necessary to transmit an executable to the client computer along with data that is processed on the client computer by the executable. This arises for example, where a web server responds to a request for flight itineraries by sending the flight itinerary data along with a JAVA™ applet for graphically displaying the data on the client. By generating part of the data while the executable is being transmitted, as required in claim 1, the method reduces delay or latency before the executable and the data arrive at the remote location. The reduction in the delay makes the method more responsive to a user.

House teaches away from this feature. Rather, as described by House, "First, control information is generated by an applet in the browser 108" (House Col. 7 lines 66-67). "Next, using this control information, a script is executed on the network server 110." (House Col. 8 lines 3-4.) Therefore, irrespective of where the applet originates, it is clear from House that the

applet is on the client and is used to generate the control information that is needed to launch a script on the server to generate the data that is sent to the client. That is, House teaches to execute the script by receipt of the control information that was produced by the client side applet. Therefore, House does not suggest generating the data at the server as the executable is being transmitted, as recited in claim 1.

Furthermore, it would not be suggested for one of ordinary skill in the art to modify House to provide this feature, since the whole point of the House reference is to produce a development tool for users of client systems. The client systems need to have the “executable” on the system so that the user can cause the executable to send the control information to the script that generates the data.

In contrast, in Appellant’s system, information (e.g., flight itineraries) is transmitted in response to a request for information (user query). While the data is being transmitted to the client in response to a request for data, that is separate from the request for information, a web page for example that contains a directive to a web browser to request the executable is transmitted to the remote location and the request for the executable is received from the web browser. The executable is transmitted to the remote location in response to the request for the executable. Thus, the feature of Claim 1 “generating the data at the server as the executable is being transmitted” provides a concomitant reduction in latency and improves responsiveness of the client and hence the experience of the user.

Tso does not supply the missing teachings or any motivation to modify House to supply this feature. The examiner merely uses Tso to teach: “processing the data by decoding and uncompressing the data for graphical presentation on a display associated with the remote location.” Appellant contends that the combination of House and Tso would not lead one to the above mentioned features of claim 1. Rather, Tso as with House, is devoid of any teachings that suggest the above features of claim 1. particularly generating the data at the server as the executable is being transmitted and transmitting the generated data to the executable at the ... client.

Appellant also contends that there is no suggestion to combine House with Tso. House is directed to a “Rapid Application Development (RAD) tool.” (House Col. 3, lines 24-27) House (at Col. 7, lines 38-54) also describes that the type of data returned to the client as:

The ButtonClick event is associated with a script on the network server 110. In this example, the script includes programming that accesses the database server through VAB-II Runtime 116 and returns data into a listbox 504 in FORM1. The data returned (script output data) from the database and network servers to the command button comprise a data string.

This data string might, for example, be "ADD DATA1 DATA2 DATA3 TO LB1". In this very simple example, this is a command string that the listbox 504 understands, and that the command button 506 now has in its possession. As illustrated, the data string comprises only a portion of an HTML page, relieving the need to transmit an entire HTML page merely to update a portion of a page. The data string may also comprise an entire HTML page. Alternatively, the data string may also comprise information used to execute a second applet in the client browser 108.

From the teachings of House it is not seen why one of ordinary skill in the art would be motivated to modify House to "decode and uncompress the data generated by the server for graphical display." Tso when combined with House is also void of any suggestion that the claimed executable transmitted from the server to the client decodes and uncompresses the data generated by the server for graphical presentation on a display associated with the remote location. Rather, Tso teaches at Col 12 lines 17-41:

According to another embodiment of the present invention, illustrated in FIG. 5, network client 12 may be "enabled," containing specialized software to support, for example, more sophisticated transcoding features than are provided by the above-described embodiments, or to perform some or all of the transcoding functions on the client side. As illustrated, network client 12 includes an HTTP local proxy 48 coupled to a client-side parser 50 which, similar to parser 22 of transcoding server 34, controls one or more client-side transcode service providers 52. Each transcode service provider 52 may be configured, for example, to transcode content before it is rendered to a user or to perform a counterpart transcoding function (e.g., decoding, decompression) with respect to a function performed by a corresponding transcode service provider 24 of transcoding server 34. As in transcoding server 34, network client 12 may include a client-side cache memory 56 managed by a client-side cache interface 54. Client-side cache interface 54 may be an already-existing facility supported by the operating system, such as WININET. Using an existing caching facility reduces the amount of software that is to be downloaded to network client 12 to implement this embodiment, and also allows other applications,

such as disconnected browsers, to share client-side cache memory 56.

Whereas, the data disclosed by House is transmitted by the server to the client and displayed or used to execute an applet, and Tso teaches a network client "containing" specialized software, neither of these references suggest "transmitting the generated data to the executable ***, the executable processing the data by decoding and uncompressing the data for graphical presentation ***". That is, Tso does not cure the deficiencies in the teachings of House, namely neither describing nor suggesting using the transmitted executable to decode and uncompress the data once the client receives the data.

Therefore, there does not exist any motivation to make the combination of references as argued by the examiner, and the alleged combination of House with Tso neither describes nor suggests at least the above features of Appellant's claim 1.

Claims 3 and 16

Claim 3 depends from claim 2 and recites that the transmitted executable is received by the client before the generated data is received and the executable causes the client computer to indicate that client computer is waiting for the data.

Claims 4 and 17

Claim 4 depends from claim 2 and recites that a first portion of the generated data is received by the client before a second portion of the generated data and the executable causes the client to indicate that the first portion has been received before the second portion is received. The examiner contends that House teaches this limitation at Col. 7 line 45-54 and Table 1 of Col. 8 (reproduced below).

This data string might, for example, be "ADD DATA1 DATA2 DATA3 TO LB1". In this very simple example, this is a command string that the listbox 504 understands, and that the command button 506 now has in its possession. As illustrated, the data string comprises only a portion of an HTML page, relieving the need to transmit an entire HTML page merely to update a portion of a page. The data string may also comprise an entire

HTML page. Alternatively, the data string may also comprise information used to execute a second applet in the client browser 108..

No such teachings are found in House. Rather, House merely teaches to return a data string to a command button. While House does teach that the data string “comprises only a portion of an HTML page, relieving the need to transmit an entire HTML page merely to update a portion of a page.”, House suggests nothing about transmitting data in multiple portions (first and second portions) and causing the client to indicate that the first portion has been received before the second portion. House also teaches to store the data string in a list box, but this teaching likewise does not suggest that a first portion of the generated data is received by the client before a second portion of the generated data and the executable causes the client to indicate that the first portion has been received before the second portion is received.

Claims 5 and 18

Claim 5 depends from claim 2 and recites that a first portion of the generated data is received by the client before a second portion of the generated data and the executable causes the client processor to process the first portion before the second portion is received.

The examiner contends that House teaches this limitation at Col. 7 line 45-54 and Table 1 of Col. 8 (reproduced above). Again, while House teaches to return a data string to a command button and that the data string “comprises a portion of an HTML page rather than an entire HTML page and to store the data string in a list box, these teachings have no relevance to the claimed limitation.

Claims 6, 11, and 19

Claim 6 depends from claim 2 and recites that a first portion of the data is generated before a second portion of the data and at least part of the first portion of the data is transmitted while the second portion is being generated.

The examiner contends that House teaches this limitation at Col. 7 line 45-54 and Table 1 of Col. 8 (reproduced above). Again, while House teaches to return a data string to a command button and that the data string “comprises a portion of an HTML page rather than an entire

HTML page and to store the data string in a list box, these teachings have no relevance to the claimed limitation.

Claims 7, 12, and 20

Claim 7 depends from claim 2 and recites that the information is transmitted in response to a request for information. House transmits control information to execute a script. House does not teach that information is transmitted in response to a request for information.

Claims 8, 13, and 21

Claim 8 depends from claim 2 and recites that the generated data is transmitted in response to a request for data separate from the request for information. House does not teach this feature. The examiner contends that the feature is shown in Figure 5B. However, the examiner does not point out what portion of Figure 5B teaches this feature. Nevertheless, claim 8 requires that the request for data is separate from the request for information. House does not show two requests nor does House show the request for data is separate from the request for information.

House shows in Figure 5B a process in which control information is generated by an applet in a browser 108 on a client 102. The control information is transmitted from the client 102 to the network server 110. A script, executed on the network server 110, uses the control information to produce the data that is output from the network server to the browser 108. Appellant contends that neither the generation of control information nor any other teaching in Figure 5B corresponds to a request. Clearly, House does not teach two requests, if House does not teach a request, but in any event, House does not suggest that the generated data is transmitted in response to a request for data separate from the request for information.

Claim 9

Claim 9 depends from claim 2 and recites that transmitting the information further comprises transmitting to the remote location a web page containing a directive to a web browser to request the executable and receiving a request for the executable from the web browser, wherein the executable is transmitted to the remote location in response to the request for the executable.

The examiner contends that House at Col. 6, lines 23-30 (reproduced below) teaches this limitation.

A Java applet is created for each scriptable control provided by the present invention. These applets may be customized by the developer. Since there is one applet per control, if there are five command buttons (for example) on one form, only one copy of the command button applet is downloaded to the client browser 108. This applet remains in the browser cache.

These teachings in House do not suggest claim 9. House does not suggest transmitting to the remote location a web page containing a directive to a web browser to request the executable. Rather, according to House the executable (Java Applet) is downloaded. No details are given as to what causes the download. Further, no mention is made of any request for the executable from the web browser nor transmitting of the executable in response to the request.

Applicant : James N. Rees
Serial No. : 09/704,221
Filed : November 1, 2000
Page : 15 of 19

Attorney's Docket No.: 09765-020001

Conclusion

Appellant submits, therefore, that Claims 1-21 are allowable over the cited art.
Therefore, the Examiner erred in rejecting Appellant's claims and should be reversed.

Respectfully submitted,

Date: 2/18/01



Denis G. Maloney
Reg. No. 29,670

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21028845.doc

Appendix of Claims

1. A method comprising:

transmitting information by:

transmitting an executable from a server to a remote location over a network, the executable processing data generated by the server;

generating the data at the server as the executable is being transmitted;

transmitting the generated data to the executable at the remote location over the network, the executable processing the data by:

decoding and uncompressing the data for graphical presentation on a display associated with the remote location; and

displaying the decoded and uncompressed data in a graphical presentation on a display device associated with the remote location.

2. The method of claim 1 wherein a client at the remote location receives the transmitted executable and the generated data and the client includes a client processor that executes the executable to process the data.

3. The method of claim 2 wherein the transmitted executable is received by the client before the generated data is received and the executable causes the client computer to indicate that client computer is waiting for the data.

4. The method of claim 2 wherein a first portion of the generated data is received by the client before a second portion of the generated data and the executable causes the client to indicate that the first portion has been received before the second portion is received.

5. The method of claim 2 wherein a first portion of the generated data is received by the client before a second portion of the generated data and the executable causes the client processor to process the first portion before the second portion is received.

6. The method of claim 1 wherein a first portion of the data is generated before a second portion of the data and at least part of the first portion of the data is transmitted while the second portion is being generated.

7. The method of claim 1 wherein the information is transmitted in response to a request for information.

8. The method of claim 7 wherein the generated data is transmitted in response to a request for data separate from the request for information.

9. The method of claim 1 wherein transmitting the information further comprises:
transmitting to the remote location a web page containing a directive to a web browser to request the executable; and
receiving a request for the executable from the web browser, wherein the executable is transmitted to the remote location in response to the request for the executable.

10. An article comprising a machine-readable medium which stores machine-executable instructions operable to cause a machine to:

transmit information by:

transmitting an executable from a server to a remote location over a network, the executable processing data generated by the server;

generating the data at the server; and

transmitting the generated data to the executable at the remote location over the network, with the executable capable of processing the data by:

decoding and uncompressing the data for graphical presentation on a display associated with the remote location.

11. The article of claim 10 wherein the instructions cause the machine to generate a first portion of the data before generating a second portion of the data and transmit at least part of the first portion of the data while generating the second portion of the data.

12. The article of claim 10 wherein the instructions cause the machine to transmit the information in response to a request for information.

13. The article of claim 10 wherein the instructions cause the machine to transmit the data in response to a request for the data separate from the request for information.

14. An apparatus comprising:
a storage system storing machine readable instructions and an executable; and
a server system that executes the computer readable instructions to respond to a request for information by:
transmitting the executable to a remote location over a network;
generating data while the executable is being transmitted; and
transmitting the generated data to the executable at the remote location over the network , the executable processing the data by:
decoding and uncompressing the data for graphical presentation on a display associated with the remote location; and
displaying the decoded and uncompressed data in a graphical presentation on a display device associated with the remote location..

15. The apparatus of claim 14 wherein the executable comprises instructions operable to cause a client processor at the remote location to process the data.

16. The apparatus of claim 15 wherein the executable is received at the remote location before the generated data is received and the executable causes the client processor to indicate that client processor is waiting for the data.

17. The apparatus of claim 15 wherein a first portion of the data is received at the remote location before a second portion of the data and the executable causes the client processor

to indicate that the first portion of the data has been received before the second portion is received.

18. The apparatus of claim 15 wherein a first portion of the generated data is received at the remote location before a second portion of the generated data and the executable causes the client processor to process the first portion of the generated data before the second portion is received.

19. The apparatus of claim 14 wherein a first portion of the data is generated before a second portion of the data and at least part of the first portion of the data is transmitted while the second portion is being generated.

20. The apparatus of claim 14 wherein the information is transmitted in response to a request for information.

21. The apparatus of claim 14 wherein the generated data is transmitted in response to a request for the data separate from the request for information.